

Arduino-eksperiment	130125	Stikord	analogWrite(), PWM, sammensat operator	
Version	2018-07-04 / HS	Niveau	Grundkursus, modul 4	p. 1/3

Det lærer du:

- Brug Arduinos analoge udgange til en lysdæmper
- Opdater en variabel med +=
- Opdag, at de "analoge" udgange ikke bare giver en jævnspænding fra sig
- Forstå, hvad *Pulse Width Modulation* er

1 – Tilslut en lysdiode til en analog udgang

"Analog" udgang skal her forstås som et ben, hvor spændingen kan varieres mere eller mindre jævnt i modsætning til en "digital" udgang, som kun kan være enten høj eller lav (5 V eller 0 V).

Alle de nummererede ben på Arduinoen (inklusive benene A0 til A5) kan bruges som **digitale** ind- og udgange. Derimod er der et begrænset antal ben, som kan bruges som **analoge** ind- og udgange. De analoge *indgange* (A0 til A5) gemmer vi til senere. De ben, der kan anvendes som analoge *udgange*, er ben 3, 5, 6, 9, 10 og 11.

Af nogle tekniske årsager, som vi ikke vil komme ind på, er ben 5 og 6 ikke helt så fleksible som de øvrige – det er nemt at undgå dem, hvis du har Arduino-ben nok i dit projekt.

Der er et lidt lignende problem med ben 3 og 11, som ikke kan bruges, hvis man også vil bruge funktionen `tone()`, som vi gør i vejledning 130135 Tone output.

For at gøre en lang historie kort: Vælg ben 9 eller 10 som analog udgang, med mindre der er god grund til andet.

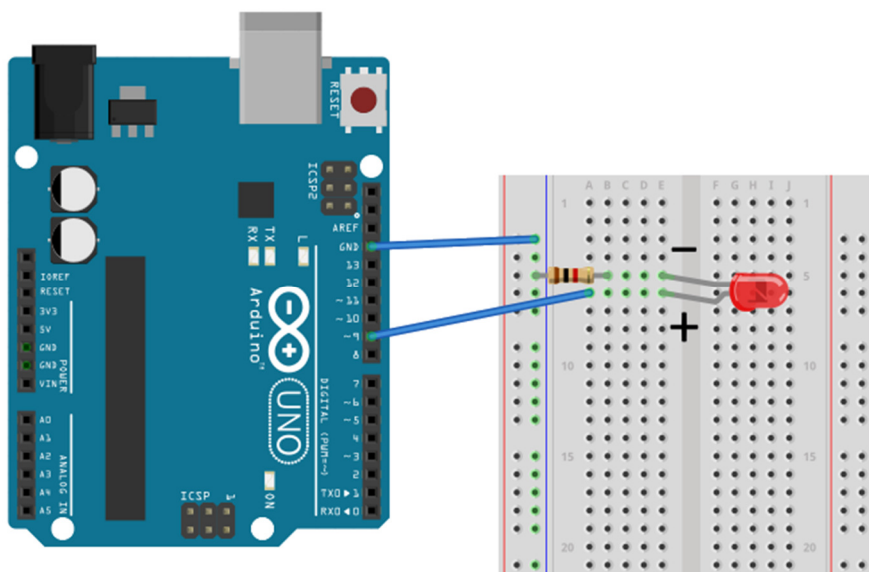
Programmet følger lige herunder. Opstillingen kan ses på tegningen nederst.

```
const byte pinOut = 9;

void setup() {
  // Intet i dette eksempel
}

int x=64;           // Tilladte værdier: 0 .. 255

void loop() {
  analogWrite(pinOut,x);
}
```



Programmet på side 1 skulle gerne få lysdioden til at lyse markant svagere, end du er vant til fra tidligere eksperimenter.

Prøv at ændre startværdien af `x` og se, hvad der sker.

Din første udfordring herunder går ud på at få lysstyrken til at svinge jævnt op og ned. Du får lidt hjælp med på vejen:

Opret en heltalsvariabel `dir`, som angiver retningen op eller ned. Giv den startværdien 1. Værdien +1 angiver "op", værdien -1 angiver "ned".

Vi kan nu lade `x` vokse eller aftage med 1 for hvert gennemløb af `loop()` ved hjælp af kommandoen

```
x += dir;
```

Den sære kombination af `+` og `=` kaldes en sammensat operator og forklares i værktøjs-boksen til højre.

Indsæt en pause på 5 – 10 millisekunder i hvert gennemløb af `loop()`, så går det passende langsomt.

Værktøj: Sammensatte operatører

Eksempel:

```
x += dir;
```

Læg værdien af `dir` til `x`.

Forklaring:

```
a += b;
```

er helt det samme som

```
a = a + b;
```

Tilsvarende,

```
a -= b;
```

er helt det samme som

```
a = a - b;
```

(Der findes yderligere sammensatte operatører, men disse er langt de almindeligste.)

Hvis du ikke gør noget, vil `x` hurtigt vokse ud over grænsen på 255. Så du skal indsætte en betingelse, som holder øje med, om `x` er ved at blive for stor, og som i så fald skifter fortegn på `dir`, så `x` begynder at aftage.

Tilsvarende: Når `x` aftager, vil den hurtigt ryge ned under nul – du må checke det med en anden betingelse og skifte `dir` tilbage til +1 igen, lige før det sker.

2 – Byg en lysdæmper

Udfordring 1

Byg programmet op omkring de ideer, der er præsenteret herover.

Lysdioden skal til at begynde med være helt slukket. Lysstyrken skal vokse jævnt op til det maksimale og skal derefter igen aftage jævnt til nul.

(Forløbet skal gentages i det uendelige.)

Der er næppe mange lysdæmpere, som helt autonomt står og cykler op og ned i intensitet. Derimod findes der flere typer, som styres med en trykknop.

Udfordring 2

Tilføj en trykknop til projektet.

Når knappen trykkes ind, skal intensiteten variere som i udfordring 1 – men lysintensiteten skal fastholdes, når knappen slippes. Der skal ikke startes fra slukket hver gang, men fra det aktuelle niveau.

Udfordring 3

Programmets opførsel skal ændres, så lysintensiteten forøges, så længe knappen er trykket ned – evt. op til maksimum. Næste gang, der trykkes på knappen, skal intensiteten falde, så længe der trykkes – evt. til nul.

Der ændres altså retning for hvert tryk – og ikke automatisk ved maksimum eller minimum.

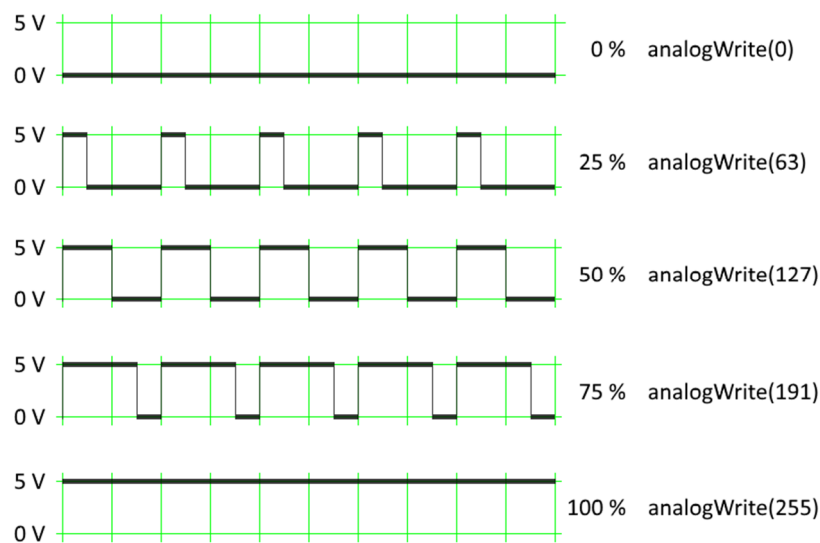
3 – PWM: Pulse Width Modulation

Når man ser lysdioden variere blødt op og ned i intensitet, skulle man tro, at spændingen på udgangsbenet er en jævnspænding, som varierer tilsvarende op og ned. Det er meget langt fra at være tilfældet!

Har du mulighed for at koble et oscilloskop til den "analoge" udgang, vil signalet afsløre sig som et digitalt signal, som svinger mellem 0 og 5 V knap 500 gange i sekundet.

Når man kalder `analogWrite()` med forskellige talværdier, styrer man i virkeligheden blot, hvor længe udgangen er højt i hver svingningsperiode. Man kan sige, at det er signalets *gennemsnitsværdi*, som varierer, når `analogWrite()` kaldes med forskellige værdier mellem 0 og 255. Denne form for signal kaldes puls-bredde moduleret, på engelsk *pulse width modulated*, forkortet *PWM*.

Den procentdel, signalet er højt, kaldes for signalets *duty cycle*. På figuren herunder ses en række eksempler på PWM-signaler med *duty cycles* mellem 0 og 100 %. Det giver god mening at fokusere på *duty cycle* som en procentsats fremfor parameteren 0 til 255 – disse værdier gælder for Arduino, men ikke nødvendigvis for andre microcontrollere.



Lysdioden står faktisk og blinker, når den tilføres et signal med en *duty cycle*, som ikke er 0 eller 100 %. Men det sker så tilpas hurtigt, at øjet ikke kan opfatte det.

Andre systemer reagerer lige så sløvt som det menneskelige øje. Det kan f.eks. fint gå at styre en elmotor med sådan et PWM-signal.

4 – Ekstra opgave

Kan overspringes, hvis du ikke har adgang til et oscilloskop (eller ikke ved, hvordan det skal bruges).

Udfordring 4 (kræver brug af oscilloskop)

Forbind et par ekstra ledningsstumper til breadboardet, så du kan klippe en oscilloskop-probe på. Probespidsen skal forbindes til ben 9, stel-klipsen skal have fat i GND på Arduinoen.

Kontrollér, at kurveformen er som beskrevet ovenfor. Mål også frekvensen af PWM-signalet.